

ESD-TR-67-547
ESTI FILE COPY

ESD-TR-67-547

ESD RECORD COPY

RETURN TO
SCIENTIFIC & TECHNICAL INFORMATION DIVISION
(ESTI), BUILDING 1211

ESD ACCESSION LIST

ESTI Call No. AL 59100

Copy No. 1 of 1 cys.

Technical Note

1967-45

G. Blustein

A Resynchronization Technique
for a Sequential Decoder

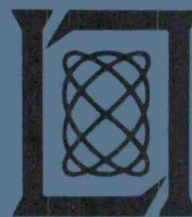
24 October 1967

Prepared under Electronic Systems Division Contract AF 19 (628)-5167 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Lexington, Massachusetts



ATD 660865

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, with the support of the U.S. Air Force under Contract AF 19(628)-5167.

This report may be reproduced to satisfy needs of U.S. Government agencies.

This document has been approved for public release and sale; its distribution is unlimited.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

A RESYNCHRONIZATION TECHNIQUE
FOR A SEQUENTIAL DECODER

G. BLUSTEIN

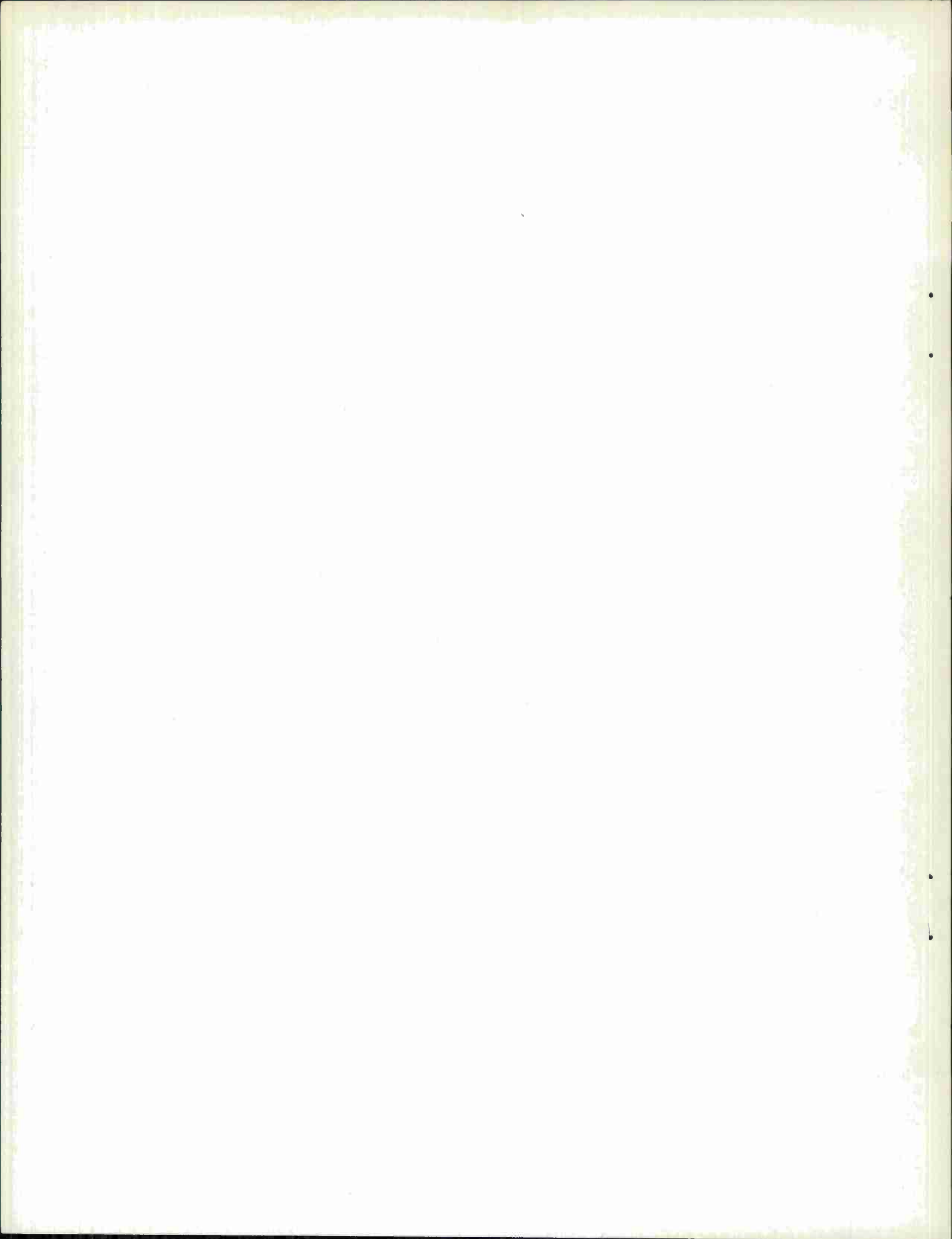
Group 62

TECHNICAL NOTE, 1967-45

24 OCTOBER 1967

LEXINGTON

MASSACHUSETTS



ABSTRACT

This report proposes a resynchronization technique for a Fano type sequential decoder. The method does not use repeat-request feedback or periodic restarting and is only employed when a buffer overflow indicates a very noisy channel condition.

A program was written to execute this algorithm for a simulated Gaussian channel for several values of E/N_0 . The results imply that this is a feasible resynchronization scheme.

Accepted for the Air Force
Franklin C. Hudson
Chief, Lincoln Laboratory Office

A RESYNCHRONIZATION TECHNIQUE FOR A SEQUENTIAL DECODER

Sequential decoding has been shown to be theoretically and practically realizable to achieve reliable communications over a number of randomly disturbed channels. However, in implementing sequential decoding the equipment model must compromise some of the characteristics of the theoretical model; essentially the infinite buffer storage and infinite encoding constraint. Truncation of the encoding constraint increases the probability of undetected error. The finite buffer permits overflow situations which causes further decoding to proceed in a haphazard fashion until resynchronization occurs. This synchronization is a restarting method for initializing the decoder to a set of known conditions at the encoder. We are here concerned with a solution to the restarting problem after a buffer overflow has occurred.

At this point we will describe the decoding algorithm in some detail. This should make the later discussions of overflow clear and give the reader an appreciation of the problem posed. The Fano decoding algorithm is employed. We assume a memoryless channel with an input alphabet I of s symbols and an output R of s random variables, each variable associated with one of the input letters. Thus the channel is defined when the resulting probability distributions of the received random variables are known for the input alphabet I . With these random variables we can compute the s values of the conditional probabilities $p(V/x)$ where V is a vector or list of the s outputs at the receiver and x takes on the values of the s symbols in I . At the receiver a list V is created at discrete time intervals Δt . The list must be kept for some time to provide for possible changes in the decoder decisions about the transmitted data.

Figure 1 is a schematic of the Fano decoder. Each radius in the inner circle is a memory unit containing a vector $V_{t+j\Delta t}$ resulting from received data at $t+j\Delta t$. Recall again that each entry in this vector is associated with one of the symbols of I . This data is never changed in executing the decoding algorithm. It is used to compute a cumulative measure associated with each decision about the transmitted information. This accumulated measure in conjunction with a set of thresholds determines when searches are made. These searches are ways of changing our past decisions so as to obtain an optimum measure for the entire sequence of decisions.



The memory as pictured in Fig. 1 is a circular buffer. Each memory cell is filled in succession and if the capacity is n memory cells then cell 1 is filled after cell n . The radial segments in the annular region, labeled H_j are the current hypotheses of the transmitted data. These segments could be considered to be part of the same memory cell containing V_j . In this sense V_j is the address of the memory unit no matter when the data was received. At regular intervals of Δt as information is received and placed into memory location V_j the data in H_j is passed on to the user.

The outer arc, C , forms a window that covers C storage elements. It is a register that contains the information bits that are the H_j hypotheses. In combination with other registers of the same length, called codes, further hypotheses are generated, each with a cumulative measure to that point. On the basis of this measure and with respect to the thresholds in effect, C , moves forward or backward. If the threshold is exceeded C shifts forward, wormlike, dropping its oldest information bits and making a new hypothesis entry at its forward end. If the threshold is not exceeded C moves back picking up a hypothesis older than its current oldest entry and tries an alternate newest hypothesis. Thus although the decoder receives information at equal intervals of time it makes decisions on a decoding at variable time intervals.

Assume that when the memory cells are filled with new vectors V_j that the H_j , although given to the sink, are left intact. Then in a search condition the decoder could find itself in a situation in which the list in cell j is unused and not directly related to the hypothesis H_j . Thus the measure for an alternate hypothesis at this point is unavailable since $p(V_j/x)$ cannot be determined. The V_j is associated with data transmitted $n\Delta t$ later than the data in which we are interested. This situation is termed a buffer overflow. Successful decoding can no longer take place until the encoder at the transmitter end and the decoder at the receiver end can start with an identical set of initial conditions.

Several methods for solving the problem have been proposed. One solution is to have periodic ending and restarting procedures. In effect this is sequential decoding by blocks. Overflow can still occur but the data loss in that event is only a block. There is in addition some loss in the effective communication rate due to informationless beginnings and tails. Another solution employs a two way feedback system with re-transmission in the event of overflow to prevent the loss of data. This method loses no information but becomes costly and complex.

The restarting procedure suggested here comes into play only when a buffer overflow occurs. Our aim is to find a set of conditions matching those at the encoder. Now the channel is assumed to be noisy such that merely accepting the most likely transmitted symbol enough times to obtain a sequence C would probably result in a sequence that differed from the transmitted one. We will therefore look for a sequence that we believe differs from the correct (transmitted) data in some small number of known positions. This is accomplished by comparing the probability $\bar{p}_k = p(V_k/\hat{H}_k)$ with some arbitrary number T_c . \hat{H}_k is that symbol of the alphabet I for which $p_k = p(V_k/x)$ is a maximum over x in I. If $\bar{p}_k \geq T_c$, \hat{H}_k is noted as True, T, else it is marked Questionable or Q. If W_c , the number of Q entries is less than some arbitrary number N_c such a sequence is called a feasible C sequence.

The next step is to find a feasible sequence of most likely symbols immediately following C whose length is F. F has associated with it parameters T_F and N_F which serve for F as T_c , and N_c serve for C.

Having found C and F which can be defined $C \oplus F = R_T$, a resynchronization trial vector, we will test our hypotheses that C is the information sequence that was transmitted. The test is simple. An attempt is made to decode for F steps with the condition that the maximum decoding measure is achieved. Such a result is interpreted as having resynchronized by having matched the encoder conditions. If the F positions cannot be so decoded those positions in C, labeled Q, take on a new permutation. A next attempt is then made to go forward over F. Each failure causes us to take a new arrangement of C until all possible configurations obtainable by varying the W_c questionable positions are exhausted. Our next step is to change F in its Q positions and start again with C having its most probable first sequence as a new starter. Lacking success after all F and C sequences have been tried the windows C and F are moved ahead some given amount and an attempt is made to find a new R_T .

Figure 2 is a logical flow chart of the proposed resynchronization technique. Figure 3 is the same as Fig. 1 except that the segment formerly marked H_j to denote sequential decoding hypotheses are marked \hat{H}_j . Now the \hat{H}_j are the most likely transmitted data on the basis of V_K alone. F, the shorter window has been described before.

A computer program was written to execute this algorithm. An M-ary modulation scheme was assumed over a gaussian channel with matched filter detection.

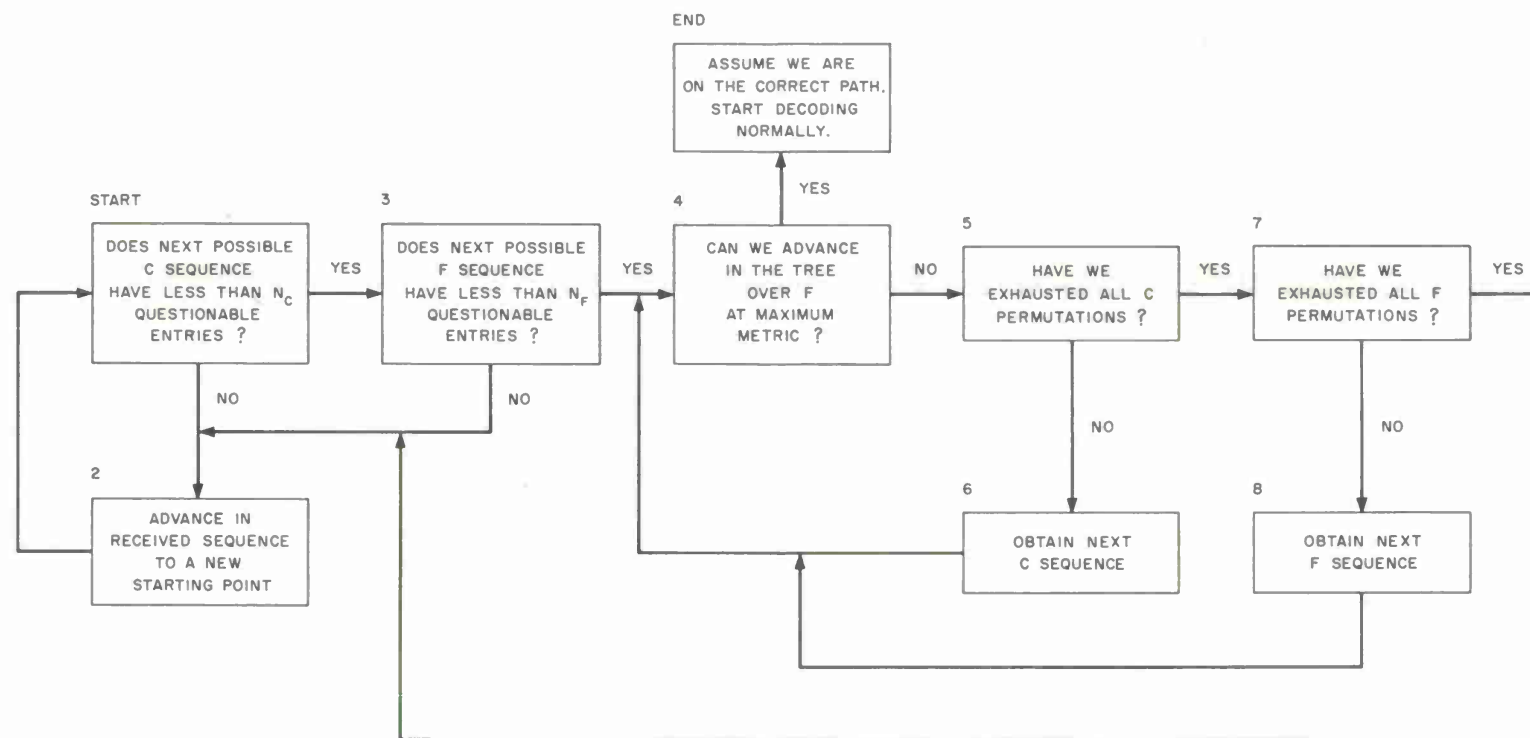


Fig. 2

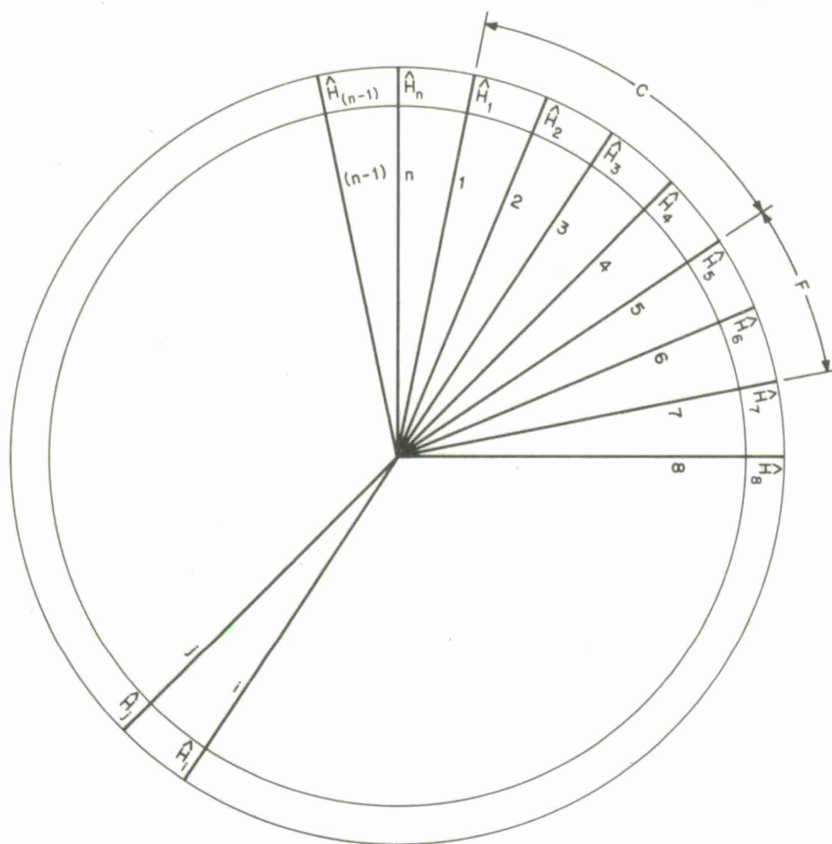


Fig. 3

The information rate was $1/3$ bit per waveform and there was one waveform per decoding step or branch. In order to simplify the program T_C and T_F were made equal and N_F was set to zero. In this way only the sequence C was varied. For the current problem the step ahead corresponded to skipping over R_T .

A number of computer runs were made for several lengths of F and varying values of T_C and N_C . E_r/N_o , the signal-to-noise power per transmitted waveform was also varied. C was fixed at 36 bits. The most interesting results are presented in Tables 1, 2 and 3. All the column headings have previously been defined except 'Memory References' and 'Resyn. False'. 'Memory References' is the total number of memory references that would have been made before a successful resynchronization on the assumption that all computation to move C forward or backward one position is done in one memory read-write cycle. This quantity does not include the memory references to find $C \oplus F = R_T$. The entry 'Resyn. False' means that the C hypothesis test indicated a successful resynchronization when in fact this was not so. All the runs consisted of 1000 transmitted symbols. The run number is merely an identifier for such a run. Each trial is a successful search for an R_T .

Table 3 presents the results of holding all parameters constant and using different random noise sequences, i.e. different samples of the same channel. The constant values of F , N_C and T_C were considered a good set. Although there are only four such runs the lack of wide variation in the results is very promising.

Better results can be expected by altering the procedure in a number of ways. The parameters T_C , T_F , N_C and F can be matched to the channel and code length C . Also there are a variety of ways to select successive trial sequences depending on likelihood and position in C . The fidelity criteria which is now either True or Questionable could be continuously valued to give a better selection of trial sequences. The hypothesis sequence can also be tested in a manner that permits some searching. All these modifications bring complications but should improve the performance of the resynchronizer.

Admittedly, the number of experiments is small. Nevertheless the method appears to be a simple workable solution to the resynchronization problem in sequential decoding.

TABLE 1

$$F_r/N_o = 3.4, R_{comp} = 1.4, \text{Transmission Rate} = .71 \times R_{comp}$$

Run	W _c	F	T _c	Total Trials	Resyn. Success	Resyn. False	Fail-ures	Memory References				
15	5	5	.75	12	1	0	11	9418				
28	7	5	.70	22	2	1	19	(184)†	50	5692		
16	7	5	.75	19	4	1	14	580	(1142)	549	1328	2157
29	7	5	.80	12	3	1	8	1423	9731	(304)	4966	
17	9	5	.75	20	2	4	14	(292)	60	650	(2926)	(4655)
								(591)				
18	5	6	.75	11	1	0	10	9458				
19	7	6	.75	18	3	0	15	547	1748	1503		
5	10	6	.75	18	2	0	16	540	2270			
7	10	6	.65	23	1	0	22	3444				
8	10	6	.85	12	0	0	12					
9	15	6	.65	23	1	0	22	3444				
10*	15	6	.75	14	2	0	12	540	2270			
14	7	6	.85	5	1	0	4	13175				
21	5	7	.75	11	1	0	10	9262				
22	7	7	.75	14	2	0	12	3644	11578			
23	9	7	.75	17	1	0	16	4595				

* 744 bits only

† () denotes a false resynchronization

TABLE 2

$$E_r/N_o = 3.0, R_{\text{comp}} = 1.2, \text{Transmission Rate} = .83 \times R_{\text{comp}}$$

Run	W _c	F	T _c	Total Trials	Resyn. Success	Resyn. False	Failures	Memory References				
30	5	5	.75	8	2	1	5	(2228)†	3744	4115		
31	7	5	.75	13	2	1	10	1574	9784	(3129)		
32	9	5	.75	17	2	2	13	850	(490)	(3512)	2287	
33	5	6	.75	7	1	0	6	5929				
34	7	6	.75	13	1	0	12	11265				
35	9	6	.75	17	1	2	14	818	(7838)	(538)		
6	10	6	.75	19	1	4	14					
36	5	7	.75	6	0	0	6	818	(4823)	(1197)	(784)	(2685)
37	7	7	.75	11	1	0	10	17238				
38	9	7	.75	13	1	1	11	(10754)	11861			

† () denotes a false resynchronization

TABLE 3

$$E_r/N_o = 3.4, R_{\text{comp}} = 1.4, \text{ Transmission Rate} = .71 \times R_{\text{comp}}$$

Four different random sequences

Run	W_c	F	T_c	Total Trials	Resyn. Success	Resyn. False	Fail-ures	Memory References				
16	7	5	.75	19	4	1	14	580	(1142)†	549	1328	2157
25	7	5	.75	17	5	1	11	51	(1994)	416	1308	83
								6281				
26	7	5	.75	16	2	1	13	423	(6104)	4485		
27	7	5	.75	17	5	2	10	2563	(2285)	2579	(1886)	160
								815	853			

† () denotes a false resynchronization

DOCUMENT CONTROL DATA - R&D		
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1. ORIGINATING ACTIVITY (Corporate author) Lincoln Laboratory, M.I.T.		2a. REPORT SECURITY CLASSIFICATION Unclassified
		2b. GROUP None
3. REPORT TITLE A Resynchronization Technique for a Sequential Decoder		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Technical Note		
5. AUTHOR(S) (Last name, first name, initial) Blustein, George		
6. REPORT DATE 24 October 1967	7a. TOTAL NO. OF PAGES 16	7b. NO. OF REFS None
8a. CONTRACT OR GRANT NO. AF 19(628)-5167	9a. ORIGINATOR'S REPORT NUMBER(S) TN 1967-45	
b. PROJECT NO. 649L		
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.	ESD-TR-67-547	
10. AVAILABILITY/LIMITATION NOTICES This document has been approved for public release and sale; its distribution is unlimited.		
11. SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY Air Force Systems Command, USAF	
13. ABSTRACT This report proposes a resynchronization technique for a Fano type sequential decoder. The method does not use repeat-request feedback or periodic restarting and is only employed when a buffer overflow indicates a very noisy channel condition. A program was written to execute this algorithm for a simulated Gaussian channel for several values of E/N_0 . The results imply that this is a feasible resynchronization scheme.		
14. KEY WORDS sequential decoding resynchronization technique decoders		

Printed by
United States Air Force
L. G. Hanscom Field
Bedford, Massachusetts

